

Transcribed from the Disk Magazine

COMPUTE!'s GAZETTE

April 1994

Disk Issue No. 04

FOR **COMMODORE** PERSONAL COMPUTER USERS



Transcription prepared for the CBM 8-Bit Magazine Index by Dr. Myles L. Skinner.
Please visit https://www.tierceron.com/commodore/magazine_index.php for
more information about the index.

GAZETTE DISK MASTHEAD

Tom Netsel, editor

Address editorial matters to the Greensboro office or via E-mail: Gaz on QuantumLink,
TomNetsel@aol.com on the Internet

Sylvia Graham, copy editor

Matthew Byrd, intern

Editorial offices:

COMPUTE's Gazette

324 W. Wendover Ave., Ste. 200

Greensboro, NC 27408

(910) 275-9809

Subscription information:

Gazette Disk

P.O. Box 3244

Harlan, IA 51593

(800) 727-6937

Call this number for any questions about your subscription, change of address, missing issues.

Advertising information:

Lucille Dennis

(707) 451-8209

Administration:

Kathy Keeton, president and COO

Bill Tynan, executive vice president operations

COMPUTE Publications Intl. Ltd

1965 Broadway

New York, NY 10023

(212) 496-6100

All material on this disk, with the exception of public domain programs featured in "PD Picks," is copyrighted 1994 by COMPUTE Publications Intl. Ltd.

Cover image created by Chris Ryan using *The Print Shop* from Brøderbund Software and originally printed on a Panasonic KX-P1180 9-pin dot-matrix printer.

Transcription prepared for the CBM 8-Bit Magazine Index and edited by Dr. Myles L. Skinner (MS). Please visit https://www.tierceron.com/commodore/magazine_index.php for more information about the index.

(The Gazette Disk masthead doesn't appear until Disk Issue No. 05, but including it here seems like the best way to give proper credit to those who deserve it. —MS)

Table of Contents

64/128 View <i>Tom Netsel</i>	3
Feedback	5
Advertising	37

Columns

D'Iversions: Ultimate Digital Intercom: Stick It in Your Ear! <i>Fred D'Ignazio</i>	7
Beginner BASIC: When and Where to GOTO <i>Larry Cotton</i>	9
Machine Language: Where to Put It? <i>Jim Butterfield</i>	13
Programmer's Page: Random Thoughts <i>David Pankhurst</i>	16
GEOS: The Paint Department <i>Steve Vander Ark</i>	20
PD Picks: MultiRaid and Putt-Putt <i>Steve Vander Ark</i>	21

Features

Review: TopDesk <i>Steve Vander Ark</i>	24
Review: GEOS Utilities Disk <i>Steve Vander Ark</i>	27
Go Ahead...Make a Page! <i>Steve Vander Ark</i>	30
Desktop Publishing Software for the Commodore <i>Steve Vander Ark</i>	34

Programs

Hyperdrive <i>Jim Wright</i>	36
Berries <i>Gus Vakalis</i>	36
Directory Manager <i>Steven Reid</i>	32
Contact—Switch On! <i>Robert Markland</i>	34
Spell Maze <i>Michael Bolin</i>	35
Autodoc 64 <i>Cameron Kaiser</i>	36
Multiraid and Putt-Putt <i>Tom Dively and Bill Britton</i>	37

64/128 VIEW

Tom Netsel

Talented programmers continue to supply the 64/128 market with exciting and innovative software. Also, here's an example of the importance of making backups of crucial disks and programs.

As you browse through this issue of *Gazette Disk*, you'll probably notice Steve Vander Ark's name appearing in several places. Readers familiar with *Gazette* will recognize Steve as the columnist who writes "GEOS" and "PD Picks." In this issue Steve provides us with those columns plus additional samples of his prose.

When he isn't searching for public domain software, Steve is often on the lookout for great GEOS utilities and applications. He found two products recently that merit consideration by GEOS users: *TopDesk* and *GEOS Utilities Disk*. You'll find his reviews of these products in the Feature section.

GEOS is popular in Germany, but not many of the applications developed there make their way to this country. *TopDesk*, a file manager program written by members of the German GEOS User's Club, is an exception. Dave Ferguson, a GEOS guru from the U.S., has translated the program into English and made it available on this side of the Atlantic. As a bonus, he's packaged *TopDesk* with several of his own programs.

Nate Fiedler is another programmer who keeps turning out GEOS products despite the mature nature of the Commodore market. Fiedler has assembled a number of his utilities and offers them under the collective title *GEOS Utilities Disk*. Look for Steve's review of this product.

While you're in the Features section of the disk, look for Steve's name again. He shares his desktop publishing expertise with us in the article, "Go Ahead...Make a Page." This article contains tips for getting the most out of *geoPublish* and DTP software in general.

Hmmm. Two columns, two reviews, and a feature article. After doing all this work for *Gazette*, I'm surprised Steve didn't submit a program or two. Actually, a number of other talented people have sent in programs for this issue. Gus Vakalis, Jim Wright, Bob Markland, Michael Bolin, Steven Reid, and Cameron Kaiser have programs on this disk, and most of them have sold programs to *Gazette* before.

Cameron Kaiser, a university student by day, has branched out as a programmer and markets his Commodore products through a company called Computer Workshops. He recently sent *Gazette* two games for review: *Mah Gong*, a cross between Mah Jongg and Solitaire, and *Sewerman*, a puzzle game with arcade action in which you try to contain toxic sludge with a complex of pipes.

These games for the 64 sell for \$14.95, plus \$2.00 shipping. They can be ordered from Computer Workshops, 3612 Birdie Drive, La Mesa, California 91941-8044. Look for reviews of these games in an upcoming issue of *Gazette Disk*. Despite his university studies and his commercial programming, Kaiser said additional *Gazette* projects are in the works. On this disk you'll find *Autodoc 64*, a documentation utility that he wrote for the 64.

Jim Wright wrote *Hyperdrive*, a program for copying disks. There have been similar programs published over the years, but I felt this was a handy one to include on the disk. It's a good idea to make backup copies of *Gazette Disk*, and this program is great for that purpose. It works with a single drive and copies one side of a disk in less than two minutes with a 1541.

If you want to make any changes or corrections to a *Gazette Disk*, it's always best to do any work on a copy. You'll notice several Bug-Swatters in this month's "Feedback." As the disk comes from the duplicator, it is write-protected, so you can't make permanent changes unless you notch the disk. I strongly recommend that you make any changes to a copy instead of the original disk. Keep the original in a safe place in case something goes wrong with the copy.

That proved to be good advice for me in compiling this issue of *Gazette Disk*. When Steve Vander Ark sent his "PD Picks," he enclosed a copy of the programs *Putt-Putt* and *Multiraid*. *Multiraid* is an educational PD program that I suspect originated in Europe. The program consists of a loader and the main program, but it wouldn't load on my computer. Steve assured me that it worked on his machine and offered to send another copy.

Since this issue was close to deadline, I decided to simply download a copy from QuantumLink. The file has an ARC suffix, indicating that it has been compressed. After the download, I uncompressed it to my working copy of *Gazette Disk*. It loaded and ran as advertised.

In putting all the programs and articles together for *Gazette Disk*, I have to move files around, rename them, and try to make them load with our menu program. *Multiraid* didn't want to cooperate. It refused to load with our menu program until I wrote a separate boot routine called MR on this disk. Finally, I had it working fine. (I wasn't so lucky with *Putt-Putt*, but that's another story. Forget the menu program; load *Putt-Putt* from BASIC.)

After adding other programs to the disk and checking them out, I went back to recheck *Multiraid* since I had to move it in the process. When I attempted to load it, it locked up again!

No problem, I had a backup disk. I make two copies as I go along, just in case there's a problem with one. As it happened, that copy wouldn't load either. It appears that any attempt to copy or move *Multiraid's* main file causes a problem. I haven't had time to find out why, but let me warn you of the fact.

Luckily, I still had the original ARC version. *Multiraid* works fine as long as I don't touch the copy. By tomorrow I hope to have this month's disk finished with everything working properly, including *Multiraid*. I'll uncompress it again and see if it won't behave this time. Thank goodness I still had the original. I'll include the ARC version of the program just in case you have any problems with it. If you do, load the compressed version and save it to a work disk. You will need an ARC utility to uncompress it, though.

It's been exciting, moving from paper to disk format. We've a whole new set of opportunities to explore and a whole new set of challenges to overcome. I could tell you about other problems that I hope we've solved, but why give away all our secrets? I wonder if Fender Tucker, the editor of *Loadstar*, has this much fun?

Feedback

Bug-swatters, comments, questions, and answers.

Bug-Swatter

There is a problem loading *Ham Net* from the January menu program. Line 550 in MENU.PROG should read GOTO 550. That does not solve the problem however, since the machine language directory reader, DIRBYT.O, apparently is corrupted.

The files for *Ham Net* are on this disk, along with an update from the author Ralph Knight. Since the new files arrived too late to be added to the program menu, they must be loaded manually from BASIC. To load Ham Net, load and run HAM with the ,8 extension. This should automatically load DIRBYT.O and NET. Be sure to check the January disk for instructions on how to use the program. Since you will have to save data to another disk with this program, it's best to make a copy of these programs on a work disk before you actually try to use them.

There is a problem loading the game *Blue Thunder* from the menu on the February 1994 *Gazette Disk*. The menu program tries to load Blue Thunder.PD, but the program is saved to disk as Blue Thunder.

The easiest way to load and run the program is to do it as you would any BASIC program. Simply insert side 2 and type LOAD "BLUE THUNDER",8 and then RUN. The program is more than 100 blocks in size, so it will take some time to load.

You can, of course, make a permanent correction that will enable you to run the program from the menu, but you'll have to cut a notch in your disk in order to write to it. This is a simple process if you have a device that's used to make these notches, but you can easily ruin a disk if you do it incorrectly. We recommend that you make a copy of your disk and make the changes on the copy.

To make the correction, simply rename BLUE THUNDER on side 2 to BLUE THUNDER.PD with the following command.

```
OPEN 15,8,15, "RE0:BLUE THUNDER.PD=BLUE THUNDER": CLOSE15
```

Some 128 users report having trouble with the February disk loading the onscreen instruction to the game *Starship Battle*. While the instructions are included as a text file that can be read from the menu, the program also offers the option of printing the instructions onscreen at the start of the game.

Apparently, when the GEOS program on that disk was added it corrupted *Starship Battle* instruction file, SB.INT. We became aware of the problem as this disk was being completed and discovered that we did not have room to include the SB.INT on this disk. Look for it on next month's *Gazette Disk*. (Due to hidden files, there are actually less than 50 blocks free on side two of this disk.)

Uninvited Wanted

Can you tell me where I can find a copy of the game *Uninvited* by ICOM? It came out in 1987 and I haven't been able to locate a copy anywhere.

MATTHEW MCBAIN
BELMONT, ON
CANADA

Try Bare Bones Software, 940 4th Avenue, Suite 222, Huntington West Virginia 25701. It had a used copy that cost \$10. It's best to call (800) 638-1123 to order and make sure that it's in stock.

SpeedRAM Count to the Rescue

I must admit that my misgivings about the diskings of *Gazette* were assuaged by a fine January issue. Good work!

I especially value *SpeedRAM Count*, using it with the Grapevine Group's 1750 REU. I am working on my second *Star Trek: The Next Generation* novel and I had been limping along far too long, switching between count and no-count versions of *SpeedScript*.

RICHARD VAN FRANK
MONTCLAIR, NJ

We're glad you liked the program. We've got a number of utilities and productivity programs coming up in the next few months. We hope you'll enjoy them and find them useful.

128 Menu Problem

The titles on the 128 Menu are legible, but the rest of the screen is garbage. Selecting a number and opting for the article brings up the text reader menu, but it is a mixture of words and graphics. Do I have a defective disk or are there problems with your programming?

L.J. HARTNETT
VACAVILLE, CA

Your disk is not defective. The problem is that you have your 128 in 80-column mode. The 128 Menu program and Textreader, programs that we've used on Gazette Disks for years, are written for 40-column mode. We regret not having informed our new disk subscribers of this fact.

If you have a question, problem, comment, or suggestion, write to:

Gazette Feedback
COMPUTE Publications
324 West Wendover Avenue, Suite 200
Greensboro, North Carolina 27408.

D'Iversions

Fred D'Ignazio

Ultimate Digital Intercom: Stick It in Your Ear!

According to a recent survey reported on several national wire services, by 1995 notebook computers will account for 60 percent of all personal computer sales. Couple that statistic with the red-hot interest in wireless personal digital communicators—beepers, cellular phones, skypagers, PDAs, wizards, Newtons, Zoomers, and the like—and what does it give you?

It gives you the epitome of a 1950s fantasy: the ultimate home-intercom system, served up as a pastel-colored, push-button, tiny techno-toy.

Kids everywhere, be on the lookout! The walkie-talkie of tomorrow is just around the corner!

A 1950s-Style Home LAN

When I was growing up in the 1950s, my folks moved us into a new home that, in retrospect, seems like an anachronism out of Victorian England. It was a sprawling four-story mansion, built at the turn of the century, with features and fixtures designed to delight the heart of any child. It had mysterious doors and secret rooms; tiny, hard-to-find back staircases; sliding-board laundry chutes; a spooky labyrinthine basement; hundreds of room-sized closets and cabinets; and roofs with nearby trees for nighttime entries, escapes, and daring feats of juvenile high-wire aerialism.

What made the house ideal from the child's perspective made it nightmarish from any adult's perspective. The home was so huge, so sprawling, and so soundproof that it was almost impossible to hear what was going on in another part of the house. I still remember my mother shouting up the stairs at the top of her lungs for us to come to dinner or to dash off to meet the school bus.

Her attempts at communicating were a losing effort. And this was true despite the fact that my mother's lung power was nearly superhuman.

It was my father's idea to install a home-intercom system into this multiroomed, soundproofed fun house. My parents agreed that it worried them that their children could disappear for hours into far-flung corners of the house and that they had no idea where we kids were or how to summon us back into the family fold.

When the new intercom was installed, my family spent days in front of the central console in the kitchen staring in awe at all the buttons and dials and at the glowing, back-lit (paper) display. It was probably my father's hope that the new intercom would function as an instrument of adult supervision and control, but it turned out otherwise.

The adults were the proud owners of the new intercom, but they were also the least competent in its operation. After many futile attempts at sending a simple message to the other end of the house, my mother finally settled on using the gizmo as a fancy radio which

piped Christmas tunes and the voices and music of Frank Sinatra, Bing Crosby, and Benny Goodman throughout the house.

My father was only slightly better. He mastered the intercom's basic controls, but his idea of a house communications network seemed to be drawn from his days in Air Force boot camp. "Now hear this! Now hear this!" he barked in a frightening reveille announcement on school days and church days and any other day in which he felt his lazy good-for-nothing children were loafing in their beds at 6:00 a.m. "Now hear this! You are all due down to the kitchen at 0630, washed, brushed, and scrubbed, and ready to eat a man-sized helping of home fries, s***-on-a-shingle, scrambled eggs, and peppers and onions!"

(I think I should mention that my father was a child of the Depression, the proud son of an Italian former priest. He owned an Italian restaurant in a nearby town. There was a gargantuan generation gap between my father and his kids. As "real American children" we had no idea where the old man got his kooky ideas about breakfast food.)

The real masters of the new intercom system, it turned out, were us children. After extensive training on the system at odd hours when our parents weren't home, we saw the true potential of the system as a strategic weapon in our unending battle against adult intelligence, interference, and control. We soon figured out how to send directed messages to any room in the house or how to broadcast our messages across the entire system. For example, using our radios, we set up mobile "guerrilla radio stations" in different rooms to block out our mother's sappy music selections. We sent the revolutionary message of rock 'n' roll to selected bedrooms and family rooms without our parents being the wiser.

Our proudest system was also our most difficult to detect. It took our mother weeks to figure out that the strange knocking sounds she heard in the kitchen were coming from the intercom and not from the 50-year-old pipes behind the kitchen walls. It was even longer before it dawned on her that the knocking sounds weren't random electronic disturbances (as the kids informed her) but devilish coded messages sent and received by children scattered through the D'Ignazio house. The results of this underground communication system were bountiful and perverse. For example, we managed to hide a baby goat in our house for almost ten days by carrying it through the kitchen in a laundry basket and keeping it in an attic closet. We then communicated its feeding instructions through the intercom using our homemade secret code.

Similarly, we functioned as a highly sophisticated spy network whenever one of the kids did something bad and was in danger of a spanking from my father. Over the years, our parents had discovered most of our hideouts, so they would look for the misbehaving child in the usual hiding places. But our system was more advanced. We used the intercom system to alert the child and his or her helpers of our parents' whereabouts as they moved through the house. This allowed our brother or sister to keep one step ahead of our parents and to double back and hide in a cubbyhole that had already been searched.

The Intercom of the Future

Now we're back to the 1990s. If Microsoft, Hewlett-Packard, and Panasonic have their way, before long we'll have home-intercom systems which boggle the imagination.

In the olden days we were stuck with crate-sized boxes and wired intercoms that cost hundreds of dollars to install. The intercoms were big, permanent, expensive, and had limited functionality.

In the 1990s, it's going to be different. Look for new home LANs that will be tiny, digital, wireless, and multimedia. Picture a little handheld intercom the size and shape of a beeper or Walkman. Punch a button and you can page your way to any room or any other unit in the house, regardless of its location. The units will have a tiny color LCD TV screen that can call up video images of any room in which you've placed a flat camera/microphone that's the size and shape of a small photograph. You'll be able to hang the camera on the wall, prop it on a table, or stash it inconspicuously on the rug in the corner of the room. The cameras will be so inexpensive you'll be able to afford dozens—or even hundreds!

Everything will be reconfigurable. Everything will be cheap and mobile. If you don't like the way you've set up the system, pick up the devices and set them down someplace else.

There will also be various sizes of intercoms, ranging from small to tiny to nearly invisible. Judging from the recent trends in blending children's fashions with children's electronics, it's likely that children's communicators will become fashion items which look a lot like earrings (styled for both girls and boys). Possibly they'll resemble clothing and fit inside baggy pants, sweaters, and jackets, in flaps built right into the pockets. The antennas and wiring undoubtedly will be threaded into the fabric of the purses, wallets, and outfits, and be colored in the same primaries and pastels as the outfits and accessories themselves.

And, don't forget: These devices aren't just communicators. They're fusion devices: a combination of computer, walkie-talkie, intercom, telephone, TV, catalog, encyclopedia, bulletin board, and message pad. You can punch in presets to digitally photograph your parents' bedroom using infrared optics on a time-elapsed sequence of one video frame every 30 seconds for a 24-hour period. Or you can dial up your family's electronic checkbook or get a summary of your parent's cable-TV viewing habits for the past year or make long-distance 900 telephone calls or convert your little intercom box into a hyper-GameBoy machine for some under-the-covers video excitement using photorealistic special effects and real digital CD-quality sound.

Intercoms of the future will make homes a virtual playground for kids. However, if you are older than a kid and if you plan to live with kids, watch out. It might be time to visit the library to check out a book on chaos theory.

Beginner BASIC

Larry Cotton

When and Where to GOTO

Last month, I mentioned that the excessive use of GOTO is one of the most common bad habits in programming. So, let's take a look at where GOTO should and should not be used.

Short programs which use GOTO are easily deciphered. When a program becomes long and complex, however, gobs of GOTOs can weave an inextricably tangled web—even for the

programmer who may be scratching his or her head trying to figure out what's happening. Occasionally, GOTO is needed in longer programs, so let's look at some examples.

GOTO is used to send control of a program to another line which can be the same as, less than, or greater than the one which contains the GOTO. If GOTO must be used, keep the hop short--within a few lines. My rule of thumb is to try to keep the target line number in view. That is, when you list the program, your GOTO and the line number it refers to should be visible on the same screen.

GOTO can be used for repetitive tasks with IF/THEN and what appears to be a strange bit of algebra. Here's what I mean.

```
10 PRINT"DEMO"  
20 T=T+1  
30 IF T < 8 THEN GOTO 10
```

T is the counter; it begins as 0 and increments one for every loop. Strangely, Commodore's BASIC syntax allows you to omit either GOTO or THEN or to put a space between GO and TO. Also, you don't have to use any spaces in the code if you prefer, although it's often easier to read if they are included.

```
30 IFT<8THEN10  
  
30 IF T < 8 THEN 10  
  
30 IF T < 8 GOTO 10  
  
30 IF T < 8 THEN GO TO 10
```

These statements are all correct, but note that the last example won't work without THEN. Of course, a more sensible (and faster) way to print "DEMO" eight times is to forgo the use of GOTO altogether, in favor of FOR-NEXT:

```
10 FOR T = 1 TO 8  
20 PRINT"DEMO"  
30 NEXT
```

GOTO can be used with INPUT.

```
10 INPUT "ENTER A NUMBER LESS THAN 8";N  
20 IF N => 8 THEN GOTO 10  
30 PRINT" [DN] THANKS!":END
```

If the user's response is equal to or greater than 8, the computer goes to line 10 again for another try. If the user never enters a number less than 8, the program persists in repeating the question ad nauseam. Here's another variation using FOR-NEXT to limit the number of times the question is asked.

```
10 FOR T = 1 TO 5  
20 INPUT "WHAT'S THE MOST POPULAR COLOR";C$  
30 IFC$="RED" THEN GOTO 60  
40 NEXT  
50 PRINT "SORRY. THE ANSWER'S RED.":END
```

```
60 PRINT: PRINT "CORRECT!"
```

This program gives the user five tries to guess the most popular color. Although it's not particularly good programming practice to jump out of a FOR-NEXT loop before it's finished, I've never encountered any major problems.

Here's another example that uses IF-THEN with GOTO to alter a program's flow, depending on the user's response.

```
10 INPUT "PICK A NUMBER FROM 1 TO 4";J
20 IF J=1 THEN 100
30 IF J=2 THEN 110
40 IF J=3 THEN 120
50 IF J=4 THEN 130
60 GOTO 10
100 PRINT"FIRST ACTION":END
110 PRINT"SECOND ACTION":END
120 PRINT"THIRD ACTION":END
130 PRINT"FOURTH ACTION":END
```

As mentioned last month, this example illustrates another programming faux pas: too many IF-THENs. However, it works if there aren't too many choices. GOTO in line 60 whips program control back to line 10 if the user's response doesn't fall within the required range. Lines 100-130 would contain appropriate corresponding actions.

A better way uses ON with GOTO. (You can use lines 100-130 in the previous example again here.)

```
10 INPUT "1, 2, 3 OR 4";J
20 ON J GOTO 100,110,120,130
60 PRINT"[UP] [UP]":GOTO10
100 PRINT"FIRST ACTION":END
110 PRINT"SECOND ACTION":END
120 PRINT"THIRD ACTION":END
130 PRINT"FOURTH ACTION":END
```

Program control goes to the line in the Jth position after the GOTO. In other words, if 2 is entered (J=2), line 20 sends control to the line number in the second position after GOTO—line 110. Line 60 now sports two cursor-ups to prevent repeating the question and contains the GOTO.

I like to use GOTO in short, temporary utilities, such as one to determine keyboard matrix values. Try this.

```
10 K = PEEK(197)
20 PRINT K
30 GOTO 10
```

Press some keys and observe the results. (This can be useful in replacing the GET statement). Stop the action with the Run-Stop key.

Probably the most common use for GOTO involves GET.

```
10 GET A$:IF A$="" THEN GOTO 10
```

Line 10 is the classic user-response line. (Again, THEN or GOTO can be omitted.) The computer just sits there, cursorless, waiting for somebody, somewhere, to press a key.

What can we do to minimize the use of GOTO? The most obvious is to use subroutines, which are accessed via GOSUB and RETURN. GOSUB, as we've learned, is similar to GOTO (but cannot be split with a space) in that it transfers control of a program to another line whose number immediately follows GOSUB.

A subroutine is usually short. It's often employed when some part of a program is accessed repeatedly, such as a part to cause a pause, to make a calculation, to sound a tone, to set up a screen, or to get some information from the user.

Recall that the last line of a subroutine must contain RETURN as its final statement. RETURN transfers control back to whatever is immediately after the GOSUB.

GOSUBs have several advantages:

- 1) They shorten programs, thus conserving memory.
- 2) They make programs more structured and thus easier to understand.
- 3) They save typing.

Enter and run this simple example which uses a subroutine for a time-delay loop.

```
10 PRINT"[CLR]"
20 PRINT"THIS IS"
30 GOSUB 100
40 PRINT "A DELAYED"
50 GOSUB 100
60 PRINT"MESSAGE":END
100 FOR T =1 TO 500: NEXT: RETURN
```

END appears at the end of line 60 to prevent the time delay from occurring again--and for another very good reason. Try removing END and observe the results. You'll see a syntax error which is all too commonly encountered when dealing with subroutines: RETURN WITHOUT GOSUB.

The computer is trying to return to a BASIC statement right after a GOSUB, but since in this case a GOSUB didn't transfer control to line 100, you get the syntax error.

Variables can be passed easily to subroutines.

```
10 I=4: GOSUB 100
20 I=6: GOSUB 100
30 END
100 K=I^2/4.2+3
110 PRINT K
120 RETURN
```

Be sure to include as much repetitive activity as possible in the subroutine. For instance, I originally wrote this with PRINT K after each GOSUB 100. But PRINT K is happier in the

subroutine (at line 110), where one statement can be used over and over. Again, remember to use `END` (line 30) to avoid `RETURN WITHOUT GOSUB` error messages.

I haven't included these programming examples on disk since they're all short and easy to type. Entering them yourself will help you learn how to use these techniques.

Machine Language

Jim Butterfield

Where to Put It?

One of the advantages of having a magazine on disk, as opposed to on paper, is that I get the latitude to supply a set of illustrative programs to go with the text. In particular, past columns often placed code at the awkward address 8192 (hex 2000). It's not an ideal address, but it works on a wide variety of machines. This time, I'll customize the code address for each of several Commodore machines; you'll find several supporting examples on this disk.

Look for the `HELLO` files for 64, 128, and PET machines. I've also included source code in `SEQ` and `PGM` files. Look for `HELLO.SRC` and `HELLO.SOURCE`, respectively.

Why is address \$2000 poor? It's poor because the code might be overrun by `BASIC` variables and also because a series of `POKE` operations is a slow way to start off a supposedly superfast program.

In machines that came out before the 128, the "middle" address of \$2000 is endangered in two ways. As new variables are created during the program run, they will be placed above the `BASIC` program. Arrays go in the same area and gobble up lots of space. Eventually, address \$2000 is reached, and the machine code will be overwritten. There's danger from the high end, too. String variables creep ever downward from high memory toward our code. The small illustrative programs that are written here are safe from these dangers. But as you write larger and larger programs, hex 2000 becomes undesirable as an address for your `ML` code.

The Best Place

There is no guaranteed best location for your machine language code, but a popular and convenient one is this: just above the `BASIC` program. This way, `BASIC` and machine language will load together (and save together, if you wish) as one unit. We will use this method here.

Putting the code behind `BASIC` has a couple of possible problems. Once you have married `BASIC` and `ML` code, you must not change the `BASIC` program's size. If `BASIC` gets larger or smaller, the machine language program will float upwards or downwards. The `ML` code may not work if it's shifted to a new location; and in any case, it's harder to call a program with `SYS` if you don't know where it has gone to.

Another problem is the one we're going to tackle now. Since `BASIC` code lives in different places on different machines, a program written for one computer won't work on another. So we must write custom programs for each.

A Simple Program

Program Hello does a simple job: It says HELLO in machine language, but it's coupled with a BASIC program that calls it. It seems like a simple task, but there are certain things we must consider carefully, especially for the 128.

If you have a 64 or earlier machine, you may skim the next section. By the way, load your machine language monitor (MLM) before you start to enter BASIC statements.

The 128 Problem

Here's the 128 difficulty. The location of BASIC can be switched around without the user even knowing that it's happened. A program using the GRAPHIC command will do this. And the relocation won't be fixed by returning to text mode with GRAPHIC 0 or GRAPHIC 5: Only GRAPHIC CLR puts BASIC back into its original place. Worse, a wedge program released by Commodore moves BASIC in an even stranger way. When our program starts, we don't know what the 128 has been doing previously.

Instead of trying to figure out what memory reconfiguration may have taken place, our program will insist that the 128 be in its default condition. We will do this by peeking address 46, in bank 15. That's something else to watch for: A prior program could have put the 128 into a BANK other than its default, bank 15. So on the 128, we might use the following code.

```
1994 BANK 15:IF PEEK(46)<>28 THEN PRINT"RESET & RELOAD!!!":NEW
2000 SYS ...
```

We'll never get to our SYS call at line 2000 if BASIC is badly situated on the 128.

Other Computers

The memory check that was recommended for the 128 is not needed as much for other machines. I have put a simple check in line 1994 of versions for other machines, peeking address 44 on the 64 and address 41 on the old PET/CBM computers. Such checks are normally not necessary. I put them in here to guard against experimenters loading programs to the wrong machines.

By the way, there is no VIC-20 version given here. Depending on memory expansion, the VIC-20 may place BASIC in any of three different locations. To detect the configuration, a more complex program would be needed, which would peek addresses 43 and 44 and use the resulting pointer to determine a valid SYS address.

The SYS Command

For line 2000, we enter an unfinished line.

```
2000 SYS XXXX
```

We want our machine language code to go behind the end of BASIC. But until we finish our BASIC coding, we don't know where that is! We can't supply the proper SYS address, but we may be reasonably sure that it will be four digits long.

Our BASIC program is complete, in an incomplete kind of way. We are ready to go to machine language coding. Call in your machine language monitor by typing **MONITOR** on the 128. On the 64 and other machines, you should already have preloaded a MLM such as Supermon.

How do you find the address where BASIC ends? On the old PET/CBM units, check the contents of address \$28/29; on the 64, VIC, or Plus-4, check addresses \$2B/2C; and on the 128, examine the contents of \$1210/1211. Type, for example, **M 1210 1211**, and you'll see something like the following.

```
>01210 3D 1C ...
```

Reverse the order of the first two bytes displayed, and you'll get \$1C3D as the first free location behind your BASIC program. Depending on the way you entered the program, the address might be slightly higher or lower, but it will be close. On 64-class machines, you might see something around \$083D, and on PET/CBM oldsters, around \$043D. The addresses may well be lower in these cases, since we won't need to write as much BASIC.

We could start our machine language program exactly at the address given, say, 1C3D, but it's easier to waste a few bytes and start at 1C40. Let's put in a simple Hello program. I'll give coding appropriate for the 128 addresses, but it will be similar on other machines.

```
A 01C40 LDX #$00
A 01C42 LDA $1C4E,X
A 01C45 JSR $FFD2
A 01C48 INX
A 01C49 CMP #$0D
A 01C4B BNE $1C42
A 01C4D RTS
A 01C4E
```

We have no code for address 1C4E, so we tap Return to take us out of assembly mode. At address 1C4E, however, we want to put our HELLO message (note that the address is used at line 1C42, above). This is data, not instructions, so we don't try to assemble it. We ask for a memory display by typing **M 1C4E 1C55**. We then move the cursor up and type over the line to make it read as follows.

```
>01C4E 48 45 4C 4C 4F 21 0D ...
```

The last value doesn't matter, since we won't be using it. Press Return to enter this data. Now, a little careful counting in hexadecimal will confirm that the first free location is \$1C55. Enter that into your end-of-BASIC pointer; on the 128, that's address \$1210.

```
>01210 55 1C ...
```

Command X to go back to BASIC. Remember that the ML code started at address \$1C40, which is 7232 in decimal. Change line 2000 to read as follows.

```
2000 SYS 7232
```

Go back into the MLM to check that you haven't accidentally moved anything when you were typing in BASIC.

If all is OK, you could save the program directly out of the MLM with the following command.

```
S "0:HELL0128",8,1C01,1C55
```

Alternatively, if you have set that end-of-BASIC pointer correctly, you could return to BASIC with X and just save the program in the usual way. Once the end-of-BASIC has been moved, the whole works saves in one shot. And it will load in one shot, too, when you bring it in.

Programmer's Page

David Pankhurst

Random Thoughts

Random is defined as "having no specific pattern." It's this lack of a pattern that makes computer games enjoyable. Randomness also allows simulations to make highly accurate predictions based on unpredictability.

The Commodore computers come with routines that create random numbers based on highly complicated formulas, but all that appears random is not necessarily so. Since the Commodore routines depend on past results, the numbers they generate are predictable, if only by another computer. Although not truly random, the numbers give every appearance of being so, and are called pseudorandom numbers.

The Eye Knows

What we usually want from a random number generator is a lack of a pattern and a good spread of values. Unless you are a professional gambler, you don't want a dice program that produces sixes half of the time or snake eyes every seventh roll. Although there are many ways to test random number generators, one simple way is to let the eye do it. If your eye can't spot a pattern, then there's a good chance that the pattern is random.

```
10 REM RANDOM NUMBER PLOTTING
15 REM LINE 50 IS INITIAL SEEDING VALUE
20 REM LINE 70 IS GENERATOR FOR X
25 DIMI(6000):CLR:B=8192: POKE53272, PEEK(53272)OR8:POKE 53265,PEEK(53265)
OR32
30 FORI=1024TO2024:POKEI,1:NEXT
40 REM -----RANDOM NUMBER SEED-----
50 X=RND(-TI/1023)
55 FORI=1TO1E37
65 REM----RANDOM NUMBER GENERATOR----
70 X=RND(1)*63999
75 REM----PLOTTING SECTION----
80 X=INT(X):Y=INT(X/8): POKEB+Y,PEEK (B+Y)OR2^(7-X+Y*8): IF PEEK(198)=0
THEN NEXT
85 POKE198,0:SYS198
```

This program, HIRES RND PLOT on the flip side of this disk, repeatedly plots random numbers on a high-resolution screen until you press a key. Line 70 multiplies the random number by 63,999 to spread the values across the pixels of the screen. Line 50 generates the seed, the initial value that all subsequent random numbers come from. The eye has a knack for seeing patterns, and a program like this lets you try out random number routines to see if they “look” random. Change lines 50 and 70 as you wish, and rerun.

Random Dos and Don'ts

BASIC's RND function is actually three functions in one: one function for positive numbers, one function for negative numbers, and one function for 0. The three work their magic in completely different ways and should be used for different purposes. Let's look at each in turn.

RND(0) gets its resulting value from the CIA (Complex Interface Adapter) chip and should be avoided like the plague. Although it is technically a random generator and not a pseudorandom one, in reality it's just not random enough. To see what I mean, run the above number-plotting program with the following line change.

```
70 X=RND(0)*63999
```

Rather than a random pattern, you'll get a series of lines. RND(0) only generates about 500 unique numbers, and this severely restricts its usefulness. This certainly is not what you'd call if you wanted random numbers from 1 to 10,000.

A positive value for RND() is the only way to go. The value that you select doesn't matter; only the sign is important. So RND(1) and RND(2) give the same result. This works by taking the result of the previous RND call (now the seed value), performing some calculations on it, and returning the result, creating a new seed value in the process. This seed value then becomes the input to the next RND call.

Sequences of random numbers vary because of the starting value, or seed. On startup, the Commodore computers place a constant into this seed value. If you don't alter it, positive RND calls will yield the same sequence every time the computer is powered up. Any game that uses this random number generator will also have the same sequence every time you play it.

To change this startup seed value, call RND with a negative value. This is the only reason to use negative calls, as the numbers they generate are not random at all.

Of course, this just exchanges one type of predictability for another, so in practice use a changing value such as TI in the call. Due to the way the RND functions work for negative numbers, you improve the results by using a very large number, or one with a fraction. Try dividing by a prime number, such as 17.

In summary, then, random numbers can be most effectively used in programs by following a few guidelines. Use one negative random number call at your program's start to get the sequence going. And for that initial call, use a varying negative number with a fraction. My personal favorite is RND(-TI/1023). From then on, use RND(1).

That's My Birthday, Too

Now that we're well on our way to nice, well-behaved random numbers, what do we use them for? One use is in simulations. Here's an example that involves birthdays. BIRTHDAY SIM, found on the flip side of this disk, runs from the menu program.

```
10 REM MATCHING BIRTHDAYS SIMULATOR
20 X=RND(-TI/91):DIM B(365):I=1:T=0:C=0
30 X=INT(RND(1)*365+1)
35 PRINT X,
40 IF B(X)=1 THEN 60
50 B(X)=1:I=I+1:GOTO 30
60 T=T+I:C=C+1:A=INT(T/C)
70 PRINT:PRINT"RUN #\"C,\"PEOPLE UNTIL MATCH:\"I,\"AVG PEOPLE (ALL RUNS)\"A
80 FOR I=1 TO 365:B(I)=0:NEXT
90 I=1:GOTO30
```

How many people would you have to poll until you found 2 with the same birthday? 300? 250? 183? Run this simulation and find out.

It loops around, randomly picking birthdays (a number from 1 to 365) at line 30), and marks that day in use. Line 40 checks to see if that day is already occupied, and if it is, we have a match. The simulation prints out the people polled and the random birthday that is generated until a match is found. The program then repeats itself.

After about 40 runs, the average remains at a stable value. This is the average number of people you would likely need to poll before you found 2 with the same birthday. Try it; the results will surprise you.

Sometimes a liability becomes a benefit. Pseudorandom generators repeat when you start with the same seed. Type in this little routine and watch the results.

```
10 X=RND(-2)
20 FOR I=1 TO 100:PRINT RND(1):NEXT
```

It always prints the same sequence. This predictability, based on that first negative seed number, leads to the following coding program. Look for RANDOM ENCODER on the flip side.

```
10 REM CODING USING A RANDOMIZING KEY
20 INPUT"CODE MESSAGE ";X$:GOSUB 100
30 POKE216,99:PRINT"ENCODED TEXT: "Z$
40 INPUT"PRESS RETURN TO RECODE";X$
50 X$=Z$:GOSUB 100
60 POKE216,99:PRINT"REVERSED TEXT: "Z$
70 END
100 REM ENCODING ROUTINE
110 INPUT"KEY NUMBER";K:X=RND(-ABS(K) /17)
120 Z$="":FOR I=1 TO LEN(X$)
130 X=ASC(MID$(X$,I,1))
140 Y=INT(RND(1)*255)
150 X=NOT(X AND Y)AND(X OR Y):REM XOR
160 Z$=Z$+CHR$(X):NEXT:RETURN
```

Subroutine 100 does the actual coding. It performs an exclusive OR function on each character of X\$ with a random number, placing the result in Z\$.

And where do we get the random numbers? Line 110 sets the seed value using a number that the user supplies via an INPUT statement. This number is our code key for encoding and decoding. Supply the same number for the decoding sequence, and you get the original strings. If a different key is supplied for decoding, the string remains garbage.

The predictability of pseudorandom numbers allows you to reverse the coding process. With truly random numbers, this would be impossible.

In closing, here are several functions that you may find handy in random number programming.

Random Number Functions

```
DEF FN R0(X)=INT(RND(1)*X)
```

Returns an integer from 0 to X-1.

```
DEF FN RX(X)=INT(RND(1)*(X+1))
```

Returns an integer from 0 to X

```
DEF FN R1(X)=INT(RND(1)*(X-1))+1
```

Returns an integer from 1 to X-1.

```
DEF FN PB(X)=(RND(1)<(X/100))
```

Returns TRUE (-1) X percent of the time.

The first three functions give ways of generating a range of integers. They differ only in whether they include X and 0 as allowable return values.

The fourth function is used for random decision making. It returns TRUE, represented by -1, X percent of the time. Suppose we'd defined the above function with X equal to 80. Look at this line of code.

```
IF FN PB(80) THEN 200
```

Repeated execution of this line would jump to line 200 80 percent of the time and continue without jumping 20 percent of the time.

The files are on disk as follows.

HIRES RND PLOT	- random number plot
BIRTHDAY SIM	- birthday simulator
RANDOM ENCODER	- secret-code code
RANDOM FUNCTIONS	- random functions

GEOS

Steve Vander Ark

The Paint Department

Productivity. That's where GEOS shines. In fact, there's no better productivity system for the Commodore than GEOS. You may recall that last month I spent my thousand words saying exactly that. The metaphor I used was that GEOS in all its various functions was a lot like an office complex: a work environment with work areas designed to handle different productivity tasks or sets of related tasks, all geared toward a finished product.

While you have that image in mind, let's poke around in the nooks and crannies and see what kinds of programs make up the GEOS environment. This month, let's check out the *geoPaint* area and see what strange and wonderful programs fill out this area of the "office."

Of course, the bigwig programs in the Paint Department are *geoPaint* and *geoCanvas*, two of the finest paint programs available for the Commodore. Working with these main programs are a variety of conversion utilities that let you import various graphic formats into the GEOS environment. These include *Graphics Grabber* (part of *DeskPack* from Berkeley), *Graphic Storm*, *MacAttack II*, *geoGIF*, *Scrap It!* (shareware or public domain and available on QuantumLink), *Import Runner*, *PaintView II* (both from RUN's *GEOS Power Pack* disks), and others.

There are plenty of accessory programs, too, tools to make working with *geoPaint* even more productive. Here you'll find *NewTools* as well as *geoStamp* and *UltiPatt* (all from Quincy Softworks, 9479 East Whitmore Avenue, Hughson, California 95326-9745).

There's a file area, too, where tons of clip art are packed for ready use. This area also opens into the Publish area, but it's here in the Paint Department that clip art is not only used in documents but edited and combined and even created from scratch. Some of the best stuff is full-page artwork from *DigiClips* (look for clip art files from Fasung Jai on Q-Link). There are other great collections, including three jam-packed disks from Springboard originally designed for *Newsroom*.

Some of these files are already in GEOS photo albums, which make them particularly handy to use. Keeping all this artwork sorted and accessible are the hardworking utilities *Scrap Grab* and *Photo Grab* (both shareware and available on Q-Link), along with that marvelous little accessory called *AutoAlbumAdd* (found on the *Collette Utilities* disk from Creative Micro Designs). *AutoAlbumAdd* automatically places the current photo scrap into the first album on the disk, saving all the time it takes to open the photo manager every time you want to save a scrap.

Here in the Paint Department you'll also find some unusual pieces of equipment, the kind of high-tech gizmos that add many options for GEOS artwork. It's here, for example, that you'll find video digitizers: *Video Digitizer* from RIO Computers; *VideoByte II*; and *VideoMate*, a cartridge that lets you digitize straight from *geoPaint*.

Next to the digitizers is HandyScanner, certainly one of the most powerful graphics tools for GEOS. Oh, neither *Video Digitizer* nor HandyScanner are specifically GEOS tools, but a conversion utility called *Handy Import* (shareware from Storm Systems, available on Q-Link) will easily import images from either of these products into GEOS.

This being the Paint Department, there have to be a lot of different colors splattered about. Of course, the 40-column section has plenty of colors to choose from, but if you look a little closer, you'll see colors in 80 columns as well! The latest release of *geoCanvas*, version 3.0, includes support for 80-column color on the 128. There's even a tool to make color changes easier. *ChromaPhobe* (CHROMA.BIN found in CompuServe's Commodore Art & Games library) will make global color changes in a *geoPaint* document with a few clicks of the mouse. That means that you can now change the background color on an entire page with one command.

There's no doubt about it: The GEOS Paint Department is certainly well equipped to handle whatever graphics chores may come along. There are many more utilities and programs that I haven't mentioned, and additional capabilities are added all the time as new programs are written or as users discover new ways to use older GEOS programs.

There is one area I haven't covered that's sort of a back corner in the Paint Department. It's a specialty area where some real magic happens. Chances are a lot of you have never heard of the programs that you'll find behind the door with the big-eared mouse on the door. It's the Animation Department, and you'll be surprised what wonders these applications can perform.

The original GEOS animation program is called *Ultimate Stage* (available on Q-Link: filenames ULT STAGE and US-DOC, both uploaded by Rob Siegel). When you double-click on this file, you'll find yourself looking at a stage with its own crude drawing program and a selection of icons that lets you cut and paste individual frames. When you've finished, you can play the whole thing back. The resolution is pretty low, and the frames don't really flip very fast. But the effect is neat to watch.

A step up in speed and resolution is *Album Animator* (available on Q-Link: filename ALBUM ANIMATOR uploaded by DEN S). This utility page-flips through a photo album to create a very nice animation sequence. You use *geoPaint* to create the frames and then paste them into the photo album. (This is an excellent place for *AutoAlbumAdd* to step in.)

But if you're really in the mood to create animation, you need *geoAnimator*. This powerful animation program lets you create full-screen presentations using photo scraps and an assortment of drawing tools. You can place filled rectangles on the screen, for example, or draw lines. The program uses a BASIC-like language for creating scripts which you can test, add to, modify, and eventually turn into stand-alone animation files. You'll find *geoAnimator* on *RUN's GEOS Companion Disk*, available from Creative Micro Designs.

PD Picks

Steve Vander Ark

MultiRaid and Putt-Putt

I get mail every so often from folks who have read my columns or articles, and it comes in a variety of ways. Sometimes it's E-mail on GEnie or QuantumLink, and sometimes it shows up in my mailbox, forwarded from *COMPUTE's* editorial office in North Carolina. I always enjoy reading the comments, even if they aren't always favorable.

The most frequently asked question about this column goes something like this: “I don’t have a modem, so where can I get more programs like the ones you talk about?” Since I received another letter with that same question in it the other day, I figured I’d answer it in this column before I get to this month’s hot shareware/public domain programs.

First of all, why not buy a modem? You can then sign up with GEnie or Q-Link, even for a few months, and download files to your heart’s content. Oh, I know that some of you may feel that you don’t know what to do online and that anyone you talk to will think that you’re a doofus or something. I went through the same thing, but the feeling lasted for about a day and a half. It didn’t take long to realize that practically everyone else online is as nice as all get out, extremely helpful, and tolerant of my missteps.

But if you are absolutely dead set against reaching into the outside world with your computer, there are a couple of other options to consider. Probably the next best source of shareware or public domain programs is your local user group. Most groups maintain libraries of programs that you can get for a low price, usually just a buck or two per disk. You may have to become a member to have access, but that’s something every computer user should do anyway (along with using a modem).

You can also buy disks of shareware and public domain files from companies that advertise in places like *Gazette*. One such outfit, called Disks O’Plenty, offers a catalog and sample disk for a couple of dollars. You just might find some treasures there. Look for other ads on the disk or in back issues of *Gazette*. You should also find several other PD and shareware sources in the Commodore buyer’s guide that’s on the March 1994 *Gazette Disk*.

Of course, since you’re reading this article, you already have a great source of some great programs: this disk! In addition to the cool stuff I find for this column, you get all those great programs that used to be offered as type-ins in the old *Gazette* magazine. (If most of you are like me, you never actually got around to typing in many of them.)

OK, let’s talk programs now. This month I told myself that I was going to get serious. I didn’t even look in the game libraries (at least at first) and headed instead into places where most of you probably don’t visit very often: educational software collections. I figured I would be able to find a couple of nice, intellectually stimulating programs that would artfully drill kids on spelling or subtraction or something. And since I’m a third grade teacher (in my spare time), I figured I’d be a great judge of what’s worth the download time.

Oh boy, did I ever find a doozy! *MultiRaid* is an arcade game that reviews multiplication facts (serious educational value) while letting you shoot things and blow them up! Let me tell you, I was in heaven.

Once I’d found that one, I was hooked on games again, wouldn’t you know it? I just had to poke around in the game libraries to see what I would find. What I found was another of those games that keep me coming back again and again. This one is called *Putt-Putt*.



MultiRaid by Tom Dively.
Q-Link filename MULTIRAID.ARC.
Uploaded by Starman35.
GENie file number 13941.
184 blocks.

This game was created with the *Shoot 'Em Up Construction Set* program. If this is an example of what that program can do, I need to buy a copy. I would love to create games that look this nice and play this smoothly. Everything about *MultiRaid* is slick and professional. The background that scrolls by has a three-dimensional look to it, the helicopter you're

flying is smoothly animated, and the weapons make spiffy little sounds as they knock out the bad guys.

Of course, this is an educational game, and you'll need to develop skill at multiplication if you hope to survive for long. As you fly along, you pass over multiplication problems embedded in the scrolling background. A few seconds after you pass a problem, you'll have to fly through one of three gates. You want to go through the one with the correct answer to the problem that you just passed. Go through the wrong gate, and you'll turn into a smudge before you can say square root.

The game gets harder as the levels increase. For one thing, there are more enemy weapons firing at you, including some nasty little tanks. The math problems also get a bit harder, too. The problems go up to 9 x 9. This game is a great way to encourage kids to study multiplication, but what will really get them playing this game are the superb graphics and excellent playability...and blowing things up.

Putt-Putt

Q-Link filename PUTT-PUTT V1.3.
Uploaded by B29Gunner.
GENie file number 13942.
52 blocks.

I love this game. As you might guess from the name, it's a miniature golf simulation, and it has plenty of sand traps and barriers to keep you putting for a long time. You control things with the keyboard, choosing both a direction and how hard to hit the ball by pressing numbers.



You have eight possible directions to hit the ball. Although there aren't any windmills or loop-the-loops to play through, it's really fun to try to come up with just the right combination of angle and velocity to make the ball ricochet off the wall and avoid the various obstacles. If you hit a sand or water trap, you lose a stroke. If you send the ball through the rough, it might bounce in any direction. If you hit the ball a bit too hard, it will skip right over

the hole or curl around the lip of the hole and roll away. In other words, it can be frustrating, just like the real thing.

But, of course, with the computer keeping track of the score, you don't have to find a way to hold onto that little pencil while you putt.

If you've come across a great public domain or shareware program that you think other Commodore users would enjoy, why not send us a copy for consideration. Send it on disk to:

Gazette PD Picks
COMPUTE Publications
324 West Wendover Avenue, Suite 200
Greensboro, North Carolina 27408

Review

Steve Vander Ark

TopDesk



Once in a while a GEOS program shows up that I immediately start using regularly. One such program is *geoWizard*; another is *Scrap Grab*. It's been a while since these products came out, but now another one has finally shown up. The name of the program is *TopDesk*, and there's no doubt about it: *TopDesk* is dynamite.

TopDesk is actually a couple of years old, and you may wonder why it's taken so long to get a review of such a great product. The reason is quite simple: *TopDesk* was written in Germany by several top programmers in the German GEOS Users Club, one of the biggest GEOS user groups in the world. It wasn't until Dave Ferguson—that guy with the dog named Dweezil—brought the program to the U.S., translated the documentation and text, and packaged it with several of his own GEOS utilities that *TopDesk* became a choice for American users.

TopDesk is a file manager program, like *deskTop* or *DualTop*, that gives you access to the various files on your disks. You can copy or erase files, for example, or run your applications. You can also select printer or input drivers and set other options that affect the running of GEOS. As far as most of these functions go, *TopDesk* is no different from most other GEOS file manager programs. *TopDesk's* claim to fame, however, is its windows.

Yep, windows. I don't mean that confusing program IBM-compatible owners use to be more like GEOS users, but there are similarities. These windows are dynamic, independent display areas for showing you what's on your disks. Oh, I realize that *deskTop's* pad area is similar, but imagine being able to resize that pad to show fewer or more icons and then being able to open another pad to show yet another set of icons from the same or different disk. Get the idea? *TopDesk* lets you have up to four windows open at once, arranged however you like on

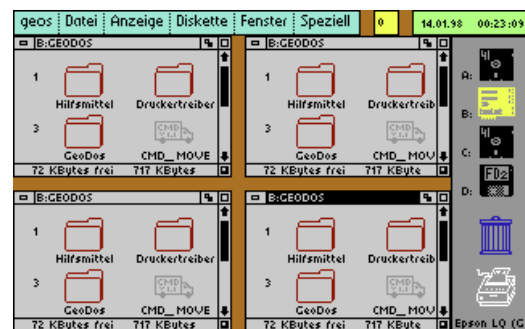
the screen. There's even a menu option to arrange the windows automatically so you can see as much as possible of each one. *TopDesk* does windows, and it's a pleasure.

You can resize the windows, but you will still be stuck with a four-icons-per-row arrangement. This means that when you make a window narrower, you can't see some of your icons. It would be nice if the display would adjust to show as many icons as possible in the space you provide. There's another menu option that can help in this case, though. You can tell *TopDesk* to display things at half size, which allows you to show the four-icon rows in half the width. This mode works just fine, but it makes the filenames almost impossible to read, even on my monochrome monitor. It's worth it, however, to have four windows open showing different areas of several disk directories all at once.

Of course, in a way this is all just cosmetic. Do you really need to see the contents of more than one disk or different areas of the same disk at a time? I'm not sure. It sure does look cool, though.

A real test of a file manager lies in the way it handles drives. First of all, it should be able to handle more than two drives. Then, it should be able to access many types of drives, both real or RAM-based. The reason this has become so important is that more and more of us are using RAM devices such as RAMLink. I like to break these monsters up into all sorts of partitions, some emulating a real drive and some not. The more variations a file manager can tolerate, the better.

TopDesk, like *DualTop*, will accept up to four drives. In order to overcome the built-in GEOS barrier against such a setup, *TopDesk* automatically swaps your drives back and forth to keep the two you're working on in the A and B slots. In other words, if you try to open a file on drive B which requires an application from drive D, the system can be set to automatically swap drive D into the A slot. While this can be a bit confusing at times, it assures you that you can always run your applications and files no matter where you have them stored. This solution to the problem is not a true fix (which would require a modification to the GEOS Kernal code), but it will keep you running. In order to use more than two drives, though, you need to have a RAM device.



So far so good, but when it comes to compatibility, *TopDesk* slips a little. It readily accepts standard 1541, 1571, and 1581 drives. I had no problem getting it to talk to a RAM 1581 partition on my *RAMLink*, either. But no matter what I tried, I couldn't get it to see my native mode partition. I tried firing up *TopDesk* from Gateway, a technique which allows *DualTop* to read a native mode partition. With *TopDesk*, all I did was confuse things. *TopDesk* told me to insert a disk called RAM 1581 into my *RAMLink* drive, and it refused to budge until I did. As near as I can tell, you can run *TopDesk* from Gateway, but you lose touch with any RAM devices. When started up from *deskTop*, though, *TopDesk* does just fine with RAM drives, but not with native mode partitions.

So what's the verdict? Is *TopDesk* worth the money, or is it just a nice example of fancy window programming for GEOS? With its limited RAM device support, *TopDesk* certainly

can't be the file manager of choice for RAMLink owners who like to take advantage of all the RAM they have with native mode partitions and subdirectories. After all, the power of being able to organize your disks in subdirectories is hard to give up.

Wait a minute. Did someone say organizing? This is without a doubt where *TopDesk* rises above all other file managers, including Gateway with its subdirectories. *TopDesk* will let you create folders on a disk (any kind of disk, not just native mode RAM partitions) where you can place whichever files you want. Unlike the subdirectories Gateway uses, these folders are really only a function of the *TopDesk* display. Files placed in one of these folders are accessible whether they're in a window showing that particular folder or not. That means that you can create a folder for fonts, clear your main directory by copying all your fonts into it, and still use those fonts in any application in any other folder.

Let me give you an example of just how neat this is. I set up my RAM 1581 partition to open with a display showing a set of main application icons—*geoPaint*, *geoWrite*, and *geoTerm*—and several folders. One of these folders holds my fonts, another holds desk accessories, and another all my *geoWrite* files. I set up another folder called Misc to hold all those oddball files that tend to accumulate, such as text scraps and preferences files.

Now, when I need to stop playing *geoTiles* and do some real work (such as finishing this review), I just open the folder for Write files (with the Games folder window still visible) and double-click on this review file. The fact that *geoWrite* isn't in the same folder makes no difference. In a few seconds I'm off and writing. When I try to open a desk accessory, I find it in the GEOS menu, even though the accessories are tucked away in their own folder.

The reason this works is simple. No files are actually moved; as far as the system is concerned, everything is where it always was. *TopDesk* is doing the organizing for you, but only with regard to the way you see things on your screen. I wish I could do this with Gateway and my big native mode partition, but with that system the files in subdirectories are isolated and accessible only from within that subdirectory.

This one feature, more than any other, makes *TopDesk* a winner. Dave Ferguson recognized the power of this system, and he added a nifty little utility to handle the one drawback to working with folders. His program Unfold/Fold will let you take everything temporarily out of folders so you can do some real organizing. You may need to actually rearrange font files, for example, to place the ones you want in the first seven slots for *geoWrite*. Once you've finished working with the directory, Unfold/Fold will put all the files back into the folders where they were.

Unfortunately, *TopDesk* doesn't get along very well with *geoWizard*. Apparently they both try to use the same area of memory in a RAM device. Someone on Q-Link suggested that a larger DACC partition on a RAMLink might fix this problem. Next time I have to reconfigure my RAMLink, I might just give it a try. I sure miss being able to task switch when I need to.

Overall, I think you'll find *TopDesk* to be a very exciting option when you pick a file manager. Now that I have my disks organized and am used to sailing through all those windows, it will be a little depressing to go back to the plain old display of *deskTop* or the scrolling list of filenames in *Gateway* and *DualTop*. If it weren't for my native mode partition and *geoWizard*, I don't think I'd go back at all!

TopDesk was written by Walter Knupe, Volker Goehrke, Ralf Bonse, and H. J. Ciprina.

Distributed by
QUINCY SOFTWARES
39479 E. Whitmore Ave.
Hughson, CA 95326-9745
\$19.95

(The *TopDesk* source code was available at <https://github.com/mist64/TopDesk> at the time I prepared this magazine—December 2022. —MS)

Review

Steve Vander Ark

GEOS Utilities Disk

Programmers like Nate Fiedler keep GEOS users like me going. No, let me rephrase that. Programmers like Fiedler keep the Commodore 64 and 128 going.

Seriously. There's no surer way to kill a computer than to stop producing software for it. That's what the software companies have done with the 64 and 128. That should mean that the 64 and 128 are history, but there are a few creative people like Fiedler saving the day.

This is especially true when it comes to GEOS. GEOS users are acutely aware of that operating system's huge potential and its many unexplored possibilities. To see the commercial companies abandon GEOS is frustrating, but talented individuals such as Fiedler, Maurice Randall, Dave Ferguson, Paul Murdaugh, and others are still creating programs that tap that great potential.

Some of these efforts result in high-power applications that deliver new tools and capabilities to GEOS, programs such as Fiedler's own *geoCanvas*. Others programs are smaller, utilities and extras that fill in some of the smaller GEOS niches. While some of these programs are small and are used by only a small number of users, all of them are welcome new additions to the world of GEOS.

The *GEOS Disk Utilities* is such a collection of programs. It's not the next big thing, by any means. It's a whole lot of neat little things. You might not have a use for all of them, but you're sure to find something that will make you say, "Hey! I've always wanted to be able to do that!"

That's the way I felt when I found the *EggTimer* program. I had always wanted a timer I could set, not for a certain time of the day, but for a certain number of minutes. This disk had two such timers on it, and I tried both. Unfortunately, only one worked in 128 mode, which is where I prefer to spend my time, but it worked like a charm.

I discovered that as long as I stayed in 128 mode, I was out of luck on a number of these utilities. Quite a few of them work their magic by fiddling with the system coding and with that of my RAM device. That wouldn't matter in 64 mode since the utilities were written for that machine, but they caused all sorts of interesting crashes on my 128.

I dug up a 64 boot disk and started over. Things worked much better. Now just about everything worked--not quite everything, though, because some of the routines apparently didn't like my RAMLink with the geoRAM poking out of the top like Devil's Tower. That was where I drew the line, though. If I dismantled my RAMlink, I'd lose all my memory and have to reconfigure. That's something I'd rather not do. So, you'll probably do as I did, pick and choose the utilities that you actually want to use. The price of the disk is low enough for you to easily get your money's worth out of it.

Here's a rundown of what's on the disk, along with a few comments. Bear in mind that I tested every file I could, sometimes with several system setups, but there are a few that I simply couldn't run. This doesn't mean they won't work, but that they wouldn't work with my system.

There are several alarm programs, including the two *EggTimer* programs I mentioned. There is also an auto-exec file called *AutoAlarm* which will set your alarm clock on boot for whichever time you set. This is handy if you like to be alerted at the same time each day to catch a show on television or something. Version 4.0 of *BlackOut*, a 64 and 128 program that protects your monitor screen by blanking it out after a specified number of minutes of inactivity, is also included.

Speaking of protection, there are several security programs on the disk. *FileLock* lets you set a password for a single file; *LockScreen* does the same for your keyboard and includes a screen blanker; *LogIn* adds password protection to your boot disk, requiring a four-character code to be entered in order to boot your system.

You can also use *DiskProtect* to remove the protection from your boot disks and master disks, converting them into normal GEOS disks. This allows you to delete files from them much more easily. You can also reverse the process, as well as add boot-disk protection to any other disk. That makes it much more difficult for anyone to delete files from it.

UnMakeGEOS converts a GEOS disk into a non-GEOS disk, although I'm not sure why anyone would ever need to actually do that.

Programmers will find uses for programs like *HeaderEditor*, which lets you change the information found in a GEOS file's header (the information you see when you select Info from the File menu), and *geoDump*, a simple memory monitor program. *DisBAM* displays a graph of the BAM, or Block Availability Map, of any disk. Its visual display is interesting but not very informative since you can't identify specific locations as being used or open.

Here are some programs that are much more useful to the average user. An excellent example is *NewSysErr*, an improved error-trapping routine that replaces those pesky SYSTEM ERROR NEAR...dialog boxes (which then lock up the system) with a soft reboot. This protects the contents of your REU.

Another program with potential is *DirPrint*, which prints out a disk directory in ASCII. That means it comes out of your printer much faster than with normal GEOS. Unfortunately, the program had no idea how to talk to my Imagewriter, preferring I'm sure the more common types of printers, such as Epson compatibles. All I got was a locked up system that even *NewSysErr* couldn't prevent.

PhotoPrint is also a printing program that's designed to print out the contents of a photo album. I've always wanted a reliable program to do this, and this one certainly works. It even liked my Imagewriter. Unfortunately, it prints one graphic per page, so my Photo Album with 12 Print Shop graphics took 12 pages, each with a single small graphic in the corner. You might find this program a bit more useful if you keep albums of larger clip art.

A couple of programs are included which modify other programs many GEOS users have. *SaveWiz* works its magic on *geoWizard*, easily the number one GEOS utility program of all time. *SaveWiz* moves the actuator code for Wizard into the REU. This means that Wizard won't be disabled in a soft reboot (such as the one *NewSysErr* now produces in lieu of a lockup).

PatchConvert modifies Convert 2.5, the utility we all use to convert files between Commodore DOS and GEOS formats. With this fix you no longer have to wade through that series of dialog boxes during a conversion; just select the file and hit OPEN. The downside of the patch is that it disables some error dialog boxes as well, but this won't really bother much.

Fiedler has also included several other helpful utilities on this disk. You'll find 64 and 128 versions of *FindFile*, a slick program which will search any disk in any drive for a filename you specify. It supports the use of the asterisk (*) as a wildcard, which helps if you can't remember exactly what a file is named.

If you have a disk with a damaged directory, you will be glad to have Redirect. This program will try to reconstruct a directory on such a disk. As the documentation states, this process is by no means a sure thing, but at least it gives you a chance for recovering lost files.

All in all, Fiedler has filled this disk with quite an impressive variety of utilities. True, a number of them have limited appeal, and if you're a GEOS 128 owner, you're often left out entirely. But the price is less than most shareware fees for one program. At \$10, you'll certainly get your money's worth of GEOS goodies on this disk.

NATHAN FIEDLER
5711 Mt. Pleasant Rd.
Bernville, PA 19506-9313

\$10.00 plus \$3.00 shipping and handling.

Go Ahead...Make a Page!

Steve Vander Ark

Tips on getting the most out of geoPublish and other desktop publishing programs.

Desktop publishing is one of the most popular ways to use a personal computer these days. From simple programs like *The Print Shop* to powerhouse applications like *geoPublish*, people are using their 64s and 128s to create cards, newsletters, signs, and anything else that can roll off a printer.

Let's face it, though, it's the software that does most of the work. With simple programs such as *The Print Shop*, DTP is actually easy. All the design choices are already made for you. All you have to do to create a card or banner is to make a series of simple choices.

Of course, to exercise your creativity, you'll have to select a program that's a bit more complicated to use. One of the best examples of a true desktop publishing program for the Commodore is *geoPublish*. This excellent program doesn't tell you where you have to put your graphics or limit you to three styles of greeting cards. With *geoPublish*, you're presented with a blank page and some very powerful and exciting creative tools. In other words, there are no limitations, just possibilities. And since *geoPublish* is part of the GEOS universe, the program can be used in conjunction with other GEOS programs to design and produce just about anything you can imagine. *geoPublish* certainly isn't simple, and by the same token, it isn't simple to learn.

If I were to give one tip to anyone who wants to learn to use *geoPublish*, it would be this: Read the manual. There are so many features and capabilities to this powerful program that you really need to read the documentation to learn to use them well. All these features, along with the GEOS connection, make *geoPublish* the most comprehensive and capable desktop publishing program around for the Commodore. The manual also provides a nice tutorial that will help you understand the basics of desktop publishing while using *geoPublish*.

There are always plenty of tips and ideas that aren't found in a manual, though. In this article I'll present a variety of tips gleaned from my own experience using *geoPublish* and other programs and from a number of books on the subject. A great many of these tips are specific to *geoPublish*, but others are general desktop publishing tips that are worth adding to the list. All of them will help you create professional-looking documents with your computer.

Plan Ahead

This is probably the biggest favor you can do for yourself when you start a *geoPublish* project. Get out a pencil and piece of paper and sketch a few ideas. Try using two columns and then three. Write in any special text and sketch where you think you'll want your graphics to go. Plan your project first on paper. Get a firm idea of what you want to accomplish.

If your project involves *geoWrite* text files, switch to that program and try setting the margins equal to the size of your project's columns. With your text loaded, you can then see how it fits the space that you have. This can help you determine the size of the font you will need and how many pages you'll have to have to get everything in. You can also avoid widows and orphans, those odd bits of text left stranded at the top or bottom of a column or page if the text doesn't quite fit the space.

Then find the pieces of clip art that you want to use. If the size of your clip art is wrong, can you find alternative pieces of artwork that don't require resizing? If you must resize your clip art, it helps to do so in *geoPaint*, where you can use pixel editing to clean up the results. When you've selected the clip art that you think you'll need, make a single photo album to hold it all. You can use Photo Grab or one of the other photo scrap handlers to put it all together efficiently.

Sketch several variations of your basic design, even if you think the first one is great. Use a ruler to determine the actual size you want for each column or graphic so you can lay the page out intelligently when you start *geoPublish*. You can use these measurements to place Master Page guidelines once you start working on your page.

I know that using pencil and paper seems a bit backward when you have all that computing power at your fingertips, but as with any job, it's best to use the right tools to get the job done. Just because you have a wonderful computer program with lots of neat features doesn't mean you should use it for everything. True, *geoPublish*'s object-based graphics can be moved and changed without any fuss. That makes creating on the fly a snap, but it's simply more efficient to plan things ahead of time.

Fonts

I can always tell when documents have been created by inexperienced desktop publishers. It's not because they don't know design tricks or how to access the best clip art. It's because they don't know how to use fonts. With GEOS you have lots of fonts to play with. Even with the program's limit of seven fonts per document, you can get wild with typefaces. Also, it doesn't take people long to figure out that by using *geoPaint* to create scraps of other fonts for snazzy headlines, you can easily beat the seven-font limit.

The problem is that too many fonts make a document look wrong. In fact, it looks amateurish and distracting. As you plan a desktop publishing project, forget about using a wild mishmash of fonts. Choose two—one for the body of text and one for headlines. You really don't need more than that.

Oh, there might be an unusual situation where throwing in a specialized font for a particular headline is what you want, but that should be the exception, not the rule. I like to use a serif font—one with those little decorative lines on the ends of the letters—for text. My favorites are Roma and Smith Corona. I often choose a sans serif—without serif—font for headlines. I like California the best for that. Each of these fonts is available in many point sizes, including megafont versions. The result is a document with a professional, unified look to it.

Now, that doesn't mean that I never spice things up a bit. Instead of throwing a whole new font into the mix, however, I use style variations. I may take the first letter of a headline and make it boldface, for example, and then set it to outline. Depending on the font, you can also use italics or a larger point size for a single letter. With a larger first letter and maybe a box from the graphics tools you can also create some very nice "dropped capitals," initial letters that are larger and offset from the rest of the text.

None of these options muddles your page with extra fonts; instead, they give your work a polished flair and keep you from looking like a rank amateur.

While we're on the subject of fonts, let me offer a few more suggestions. First of all, use megafonts for your headlines and other special text whenever you can. These fonts will give you the best-looking results for larger text.

Second, always use existing point sizes whenever possible. They will look much better than resized versions when printed. If you must resize your text, try to use multiples of the existing sizes. If you need the font Bancroft for a headline, it comes no larger than 24 points. For a larger headline, resize it to 36 points, one-and-a-half times the original size, or 48 points, two times the original size. Multiples such as these require less guessing on the part of the program as it tries to re-create a font in a new size.

Object Placement Tools

geoPublish has several features which can help you place objects carefully. You can really help yourself by placing Master Page guidelines where you want to locate various objects. Once you've placed these guidelines, you can use the Snap To feature in Page Graphics mode to make your cursor automatically fasten itself to those guides. With Snap To turned on, you can line up all your graphics and text objects easily.

A similar feature called Ratchet makes the cursor move in preset jumps. This feature makes it easy to create a series of identical lines. Draw one line and then use the Ratchet to move the cursor to create another and another. Snap To and Ratchet make creating pages faster and a lot less frustrating.

White Space

Another way that amateur desktop publishers give themselves away is by cramming too much onto a page. When you plan your pages, remember that a generous helping of empty white space is essential to making the page readable and pleasant to look at. Notice the pages in many quality magazines. See the white space along the margin? By placing that space there, the editor makes this page a little less crowded and more open. It also provides a place to put "pull out" text, that snippet of text which can draw your eye and whet your appetite to read the rest of the article.

Plan white space into your document right from the start; it's far too important an element of the overall design to leave to chance.

Notice also that, with the exception of advertisements, all the text pages in magazines usually have that same general layout: three columns with an area of white space on the outside and a header area on top. This illustrates another good idea for laying out multiple-page documents: Strive for a consistent look from page to page.

Utilities

geoPublish is a complete program. It even includes a stripped-down version of *geoWrite* to make editing text easier. There are, however, a few utilities you might want to have that can make your geoPublishing project run even smoother. While *geoWrite* and *geoPaint* can hardly be considered mere utility programs, they are powerful additions to your *geoPublish* arsenal. I recommend that you add them to your publishing arsenal. Later, I'll mention some other excellent programs that you might want to acquire.

Your text files will almost certainly start out as *geoWrite* documents. As you work on these documents, you can use a few strategies to make it easier to use them later with *geoPublish*. If you have a graphic you want to be part of a text block, for example, you can paste it into your *geoWrite* text file and then pour it into a text region in *geoPublish* along with the text. Of course, when you get to *geoPublish*, you have to be sure to create a text region for that file wide enough to hold the graphic.

Do yourself another favor when you're in *geoWrite*, use the font and point size that you'll use for your *geoPublish* document. As I mentioned before, it helps to set your margins as close as possible to the size of your final document.

You can use *geoPaint* to manipulate your graphics in ways *geoPublish* can't. Because you're using an entirely different set of rules in *geoPaint*, you can edit each individual dot of your artwork.

If you want to do this with a section of your *geoPublish* page, you can convert things into *geoPaint* format and edit them in *geoPaint*. To edit the entire page, you can use *Paint Pages* to "print" your *geoPublish* page into a *geoPaint* page. Then, if you want to convert the whole thing back into *geoPublish*, you'll have to use *Scrap Can* or *Scrap It* to turn the whole page into a photo scrap that you can paste onto a *geoPublish* page. Once you've done this, however, the individual graphic objects and even the text on the page can no longer be edited using *geoPublish's* tools, so save this step for last.

If, on the other hand, you only want to fine-tune one small section of your page, you can use a program like *Screen Grabber* to copy any portion of the visible screen to a photo scrap. This scrap can be edited in *geoPaint* and then pasted back into *geoPublish*. Once again, however, the various objects which made up this graphic can't then be edited with *geoPublish* tools.

With all this jumping between programs, you will appreciate having *geoWizard*. From *geoWizard's* menu, which you can call up from just about anywhere in any program, you can start up any other program, use it, and then exit back out to where you left off. In other words, you can be working in *geoPublish*, copy a section of the page to a photo scrap using *Screen Grabber*, use *geoWizard* to switch quickly to *geoPaint* to paste, edit, and copy the graphic back to the photo scrap, and then return instantly to *geoPublish*. From there, you can paste the now-edited scrap back onto the page. The amount of time that *geoWizard* can save you is enormous.

Printouts

The quality of your printout can make even the best planned and designed document look feeble, but there are ways to improve the looks of your finished product. Probably the most important tool is the printer driver that you are using. If you can find a double-strike driver for your printer, use it. There are drivers that offer three, four, or even six passes during printouts. These can sometimes eliminate the jagginess from your type.

There are also drivers that will interpolate as they print, meaning that they'll do their best to fill in extra dots to smooth things out. LaserMatrix drivers are a good example of this type.

You might also try a driver that prints a reduced image. There are several of these for Epson-compatible printers that achieve beautiful results by reducing everything. All the jaggies get reduced as well.

You can use an enlarging copier then to bring your document back to full size, while keeping the improved image quality. Since many desktop publishing jobs are destined for a copier anyway, this extra step in creating your finished product is often worth the effort.

The best printer drivers I've found for my 24-pin Epson printer are those written by George Wells. His drivers include a draft-quality one for quick checking of how a page looks and also an excellent reducing one which compresses the printout slightly with stunning results. Look for these and other alternative drivers on QuantumLink and GENie.

Don't ignore the possibilities of using something other than regular computer paper for your printout. A more porous paper will tend to make the ink blur slightly. This blurring can eliminate some of the jaggies at the same time. I like to use construction paper for signs. It provides me with larger sizes and a porous surface, plus a nice variety of colors.

I started this article by suggesting that reading the manual is the best way to learn to use *geoPublish* effectively. I'll finish by telling you the second best way: practice. I know that sounds hokey, but it's true. You don't get good at basketball just by buying \$200 shoes; you have to put them on and play a lot. The same is true for desktop publishing. You may have a snazzy DTP program with *geoPublish*, but you still have to get in there and use it. You have to use it repeatedly if you want to become proficient. Of course, employing these tips and ideas won't hurt, either.

Desktop Publishing Software for the Commodore

Steve Vander Ark

A list of DTP titles and where to buy them.

Finding good desktop publishing software for the 64 or 128 can be a bit of a chore these days. Sometimes you can find good buys on used software, which is a great way to pick up DTP titles that have been discontinued. For used software, check with the following distributor.

Bare Bones Software

940 4th Ave., Ste. 222
Huntington, WV 25701
(800) 638-1123

The following list of programs should still be available commercially. You will find many of these programs in the catalogs of Tenex and Software Support International. The entire GEOS line of programs is now available through Creative Micro Designs, which also sells the GEOS Companion and Power Pack disks.

Because many titles are no longer produced by their original companies, look for them at independent distributors.

GEOS, *geoPublish*, *Paint Pages*, *geoWrite*, *geoPaint*, *Scrap Can* (on the *geoCanvas* disk), *geoWizard* (on the *Collette Utilities* disk)

Creative Micro Designs

P.O. Box 646

East Longmeadow, MA 01028

(413) 525-0023

Call or write for current price list.

The New Print Shop—\$44.95

Brøderbund Software

500 Redwood Blvd.

Novato, CA 94947

Swift Desktop Publisher

Diskart GEOS clip art collections

Tenex Computer Express

56800 Magnetic Dr.

Mishawaka, IL 46545

(219) 259-7051

Personal Newsletter for GEOS

SOFTSYNC

Newsroom and *Newsroom* clip art disks

SPRINGBOARD

Outrageous Pages

Paperclip Publisher

Batteries Included

Printmaster Plus and art gallery disks

Unison World

All available from

Software Support International

2700 N.E. Anderson Rd., Ste. A-10

Vancouver, WA 98661

(800) 356-1179

Screen Grabber by Richard A. Rardin

COMPUTE'a Publications

Gazette (January 1990)—\$3.50

324 W. Wendover Ave., Ste. 200

Greensboro, NC 27408

(910) 275-9809

Scrap It

QuantumLink

Filename SCRAP IT; uploaded by TerryV7.

Programs

Hyperdrive

Jim Wright

A fast disk-copy program for a single drive and a 64.

Have you ever lost or damaged a disk that contained the only copy of an important program or file? If you have, then you know how important it is to have a backup of your disks.

The problem is that most copy programs are slow, and the ones that are fast are expensive. It's frustrating enough to make you ignore making those necessary backups.

This is where *Hyperdrive* comes in. With it, you can copy a disk in less than two minutes with a single 1541 disk drive. (This is an excellent program for making backup copies of your *Gazette Disks*.)

Hyperdrive is written entirely in machine language, but it loads and runs like a BASIC program. After it runs, you will be prompted to insert the source disk into the disk drive. The source disk is the disk that you want to copy. Place this disk in the drive and press Return.

The screen will flash with each sector on the disk being read and will then revert back to the original screen. It will report (but not duplicate) any errors found on the disk and copy the underlying data. This program won't duplicate copy-protected games.

A new message on the screen then instructs you to insert the backup disk into the drive and press Return. The backup disk is the copy that will be made of the source disk.

As it writes, the program will automatically format the disk, erasing all previous information that may have been on the disk. After this is done, *Hyperdrive* will alert you of any errors encountered while writing. If there are any errors reported, it probably means that your disk is physically damaged in some way.

After the program goes through this process three times, your copy will be complete, and the program will restart. At this point, you can make another copy or copy any other disks that need backing up.

Jim Wright lives in Wilderville, Oregon.

Berries

Gus Vakalis

An arcade-style game for the 64.

You are a hungry blue bird alone in a cat-infested garden. You want to grab a bite to eat, but the cats have the same idea! The object of this game is simple: Eat the berries. *Berries* is written entirely in machine language, but it loads and runs like a BASIC program. Use a joystick plugged into port 2 to control your bird as it flies around the garden. Note that you cannot stop flying but must remain in constant motion.

Mealtime

At the start of the game and at each new level, you'll have 20 berries randomly scattered about the garden. You can eat the berries now, but they're only worth 10 points apiece. If you wait, the berries will grow larger and be worth more points.

The berries go through five stages of growth, being worth 10, 20, 50, and 100 points. In their final stage, the berries decay and become poisonous. This is indicated by their brown color.



Obstacles

Poisonous berries are not the only hazard that you'll face in this game. You'll also have to avoid flying into the walls of the garden. You'll have to avoid a few beehives that are scattered about, and you especially have to watch out for the cats!

On the first level of the game you'll have to face only one cat and one beehive. As levels increase, more cats and beehives will be brought into play until a total of ten cats and ten hives share the garden with you.

The cats spend most of their time randomly roaming around the garden until you come within their range. Then you'll have a lot of tail shaking to do. Just to keep things interesting as you progress through the levels, each new cat will have a greater range than the previous one.

Cat number 1, for instance, has a range of only two spaces making it easy to avoid. The range increases by one space for each succeeding cat, giving cat number 10 a range of 11 spaces. This one will give you a run for your money!

Scoring

You begin the game with three lives and earn an extra life for every 5000 points. The riper the berries the more points they're worth. If you eat all 20 berries at their peak of ripeness, you'll receive a 2500 point bonus.

You can pause and resume the game at any time by pressing the Run/Stop key. You can reset the game to the first level by tapping the Restore key, and you can exit to BASIC by pressing the Q key.

At the top of the screen are four readouts indicating how many lives you have remaining, your current score, how many berries you've eaten on this level, and the level that you're currently playing.

Gus Vakalis lives in Jamaica, New York.

Directory Manager

Steven Reid

A directory manager utility for the 64.

Directory Manager is a program for the 64 that allows you to arrange a disk directory in any order you wish. Just select the file you wish to move and, using the cursor keys, move it to the desired place in the directory. This allows you to arrange files into meaningful groups and is an important part of proper disk management.

Directory Manager is written in BASIC, so use *The Automatic Proofreader* to check your typing. It's very important that this program is entered correctly, because a single error could ruin an entire disk. You might want to try the program on an unimportant disk until you are sure that there are no typos.

Using the Program

When you run *Directory Manager*, which is written in BASIC, the directory of the disk in drive 8 will be read into memory. A few seconds later, you will see the first 23 filenames of the directory. You will also notice a pointer on the first filename. You can move this pointer with the cursor up/down key, as well as the Home key. If you have more than 23 files on that disk, you can move to the next or previous screen by moving the pointer past the first or last filename. You can also move to the next group by pressing the + or - keys.

Place the cursor next to a file that you wish to move and press Return. The filename will then be highlighted. Use the cursor up/down key to move the highlighted filename to wherever you want it to appear in the directory and then press Return to place it. Continue picking up and dropping filenames until you have a well-ordered directory.

When all the programs and files are in the desired order, press W to write the directory back to disk. You will be asked if you are sure you want to write the directory. If you really are sure, press Y, and the new directory will write to disk. To view the new results, type LOAD"\$",8 and LIST, just as you usually would.

Other Options

In the upper right corner of the screen is a list of options. Here is an explanation of each.

Read Directory. Pressing R will read a new directory into memory. Use this option when you want to begin work on a new disk.

Write Directory. As mentioned before, press W to write the directory in memory back to the disk in drive 8. Be careful, however; if you write the directory to the wrong disk, you'll lose everything on that disk.

Edit Filename. Press E to edit a filename, even difficult names like those containing a comma. You can cursor left and right through the name, as well as insert and delete characters. If you want to start fresh, press the Home key to clear the filename. When finished, press Return. If you don't want to change the filename, clear it, and then press Return.

Sort Director. Press S to sort the directory into alphabetical order. The sort is usually very fast. My work disk with 39 files sorts in less than ten seconds. Larger directories will take considerably longer.

Quit Program. Press Q to return to BASIC. Directory Manager is still in memory, and you can usually enter CONT to return to the program.

Steven Reid lives in Squamish, British Columbia, Canada.

Contact—Switch On!

Robert Markland

A math game for the 64.

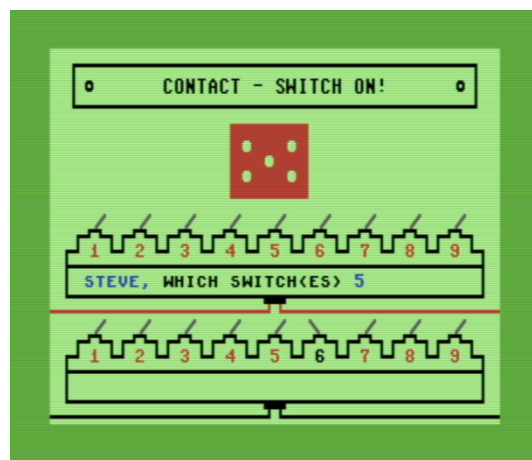
Man your switch panel—roll the dice—and be the first player to complete the circuit. *Contact—Switch On!* is an entertaining educational game that combines counting and addition skills, strategy, and a little luck. Two players may compete, or one can challenge the computer.

Contact—Switch On! consists of a loader and the main body of the game, both written in BASIC. The special characters used by the game are written in machine language. This file, SWITCH CHR5, loads automatically when you run the game.

Rules of the Game

Contact—Switch On! is suitable for small children, but the subtle aspects of strategy and a knowledge of the laws of probability make it challenging for adults as well. The rules of the game are very simple, and the addition only involves sums up to 12.

The object of the game is to turn on all nine switches of an electrical circuit. To turn on a switch, you must match its number by rolling dice.



To begin the game, select one or two players. At the prompts, enter the name of each player. A computer player called Chip will be the opponent if only one human is playing. The computer randomly determines who will go first, and play alternates until a winner is declared.

At the start of each turn, decide if you will roll one die or two dice. Usually it's advisable to roll both dice to take advantage of the greater possible number combinations. If only switch 1 remains off, you will obviously roll a single die in an effort to roll a 1.

Enter the number or numbers of any off switches that equal the number(s) shown on the dice. For example, if the total number on the dice is 5, you may elect to turn on switch number 5, switches number 1 and 4, or switches 2 and 3. (It isn't necessary to use the same numbers that appear on each die, so long as the totals are equal.) Enter your chosen numbers in any order

without separators. Thus 5, 14, 41, 23, or 32 are all valid combinations in the previous example. The computer checks for accuracy and will ask you to try again if the number(s) you enter don't equal the total shown on the dice.

Players may not skip turns and must turn switches on if they are able to do so. In the event there are no possible combinations of off switches, the computer will inform the current player, and the turn will pass to the other player.

If you tire of a game and wish to resign, type X when asked for the number of dice or your switch selections.

The game ends when one player succeeds in turning on all nine switches. Following the end of the game, you will be asked if you want to play again and, if so, if you want the same or different players.

Robert Markland lives in Newcastle, Wyoming.

Spell Maze

Michael Bolin

A spelling game for the 64.



Spell Maze is an educational game that's designed to help a student with spelling. Among other features, the game has several levels of words to spell arranged in varying degrees of difficulty.

Spell Maze is written in machine language, but it can be loaded and run like a BASIC program. Play the game with a joystick plugged into port 2.

When you see the opening menu, use the joystick to select a difficulty level. You can also select **Quit** if you decide not to play.

After you select a playing level, the word to be spelled will appear on the screen. Study it and press any key when you are ready to play.

When you press a key, the letters of the chosen word will be scattered randomly among a maze. In addition to the scattered letters in this maze, you'll also discover several keys, locked doors, and a sword. To make things challenging, you'll be able to see only a 3 x 3 grid around your man when he's in the maze. Use the joystick to move your man throughout the maze. The object of the game is to maneuver through the maze searching for the letters that spell your word. You must pick up the letters in the proper order.

Your current number of keys; the letters of the word found so far; and the sword, if you have it, will be displayed at the top of the screen. If you encounter a letter that you want, a key, or the sword, move your man on top of it and press the fire button to collect it.

You can pass through locked doors (represented by stars) only if you have at least one key in your possession. Also, to spell the word correctly, you must pick up the letters in order. You can see which letters you have already found at the top of the screen. If you pick up the wrong letter or if you finish the word, a message will inform you of this. Press any key to return to the main menu.

On rare occasions, you will be attacked by a monster. If you have the sword in your possession at this time, you will defeat the monster. Otherwise, your keys, the sword, and all the letters of your word will be scattered throughout the maze again, and you will be returned to your home.

You may press H at any time during the game, and the chosen word will be displayed at the top of the screen. Press a key to continue playing.

Pressing f1 will scatter everything in the maze. This option is helpful should you be trapped at a locked door without a key. Press Q anytime to return to the menu.

The words in the expert level are considerably more difficult than the intermediate words. To get a word that is not too easy or too hard, you may have to restart the game.

You can change the color of the yellow light around your man by loading *Spell Maze*. Before running the game, type POKE 2094,color. Color is any number between 0 and 15. If you are not familiar with these colors, a list of them can be found in the *Commodore 64 Programmer's Guide*. Just be sure that you don't select black, or you won't be able to see anything.

Michael Bolin lives in Rockwell, Iowa.

Autodoc 64

Cameron Kaiser

A documentation utility for the 64.

Many of my utilities come from *Gazette*, and that fact has caused me some problems. Since I have a very bad memory, I often forget how to use these utilities. So I have to go rooting through my magazines to find the documentation that explains how to activate and employ them.

After a while, I thought that it would be nice to have the documentation right there with the utility. That's why I wrote *AutoDoc*, to make the utilities and programs I use self-documenting. Now, the information is right at my fingertips.

AutoDoc is a specialized wedge that clips into the 64's LOAD vector. When a program is loaded, *AutoDoc* is invoked. If there is a documentation file available for this program, *AutoDoc* displays it and then loads the program, all in one fell swoop. Best of all, *AutoDoc* is transparent to most utilities and fastloaders and won't choke on programs that don't carry documentation files.

AutoDoc consists of three portions: a machine language control routine, a BASIC installer, and a BASIC screen maker. The ML file is AUTODOC.ML, and it gets loaded by the installation routine. The ML loading and ending addresses are 9F00 and 9F97, respectively.

The two BASIC routines are the installer, AUTODOC.BOOT, and the screen maker, AUTODOC.WRT.

AutoDoc documents itself when you run it, providing the addresses for controlling it. It then installs the main program and then clears it from memory. You should use AUTODOC.BOOT initially to load instead of loading the ML file and using the SYS command to activate it.

The install program protects *AutoDoc* so BASIC won't digest *AutoDoc* and crash when it encounters a LOAD command. After installing *AutoDoc* with the install program, you can use the SYS commands afterward with no problem.

Using the Program

Before you can actually put *AutoDoc* to work, you must create a documentation file. To do this, load AUTOOC.WRT. When you run it, it prompts for the program's filename that the documentation file will link to. Enter that program name.

DO NOT ADD THE "I." The program will add it for you. You will then get a blank screen. You can cursor about in it, using any letters or graphic characters that you need to provide yourself with enough information to use the program.

Note that you can't press Return to skip lines. That aborts the program. Use Shift-Return instead. When you've finished, press Return and the program will save the screen for you (without color, however). The program adds a prefix (I.) to the filename to indicate that it's a documentation file for *AutoDoc*. You can even document AUTODOC.WRT if you like.

Using Autodoc

When you want to run a utility or program that has *AutoDoc* documentation, first boot AutoDoc. To use it, simply load a program as you normally would. If a documentation file is present, you will see the following.

```
SEARCHING FOR I.PROGRAM NAME
LOADING
```

The documentation will appear while *AutoDoc* loads the actual program.

If there isn't a documentation file, you will see the following.

```
SEARCHING FOR I.PROGRAM NAME
SEARCHING FOR PROGRAM NAME
LOADING
READY
```

Notice that *AutoDoc* doesn't cause any errors if you run it on a program that doesn't have a documentation file with the I. prefix. Instead, *AutoDoc* smoothly moves to the actual program.

A Few Notes

Unfortunately, not all utilities are perfect, and *AutoDoc* is not an exception. First of all, you can't install *AutoDoc* twice. That is, don't turn it on twice in a row. That makes the **LOAD** command lock up the computer. If you find out you've done this, try SYS 65418 to restore everything.

Second, while *AutoDoc* has been intentionally made as transparent as possible—it is transparent to BASIC and occupies a rarely used block of memory—there will be certain utilities that will refuse to coexist with it. There will also be programs that will erase *AutoDoc* by accident and then crash while attempting to access the disk drive. You simply have to be aware of these programs and then disable *AutoDoc* before running them. What may help, especially in the case of fastloaders, is to disable *AutoDoc*, install the program, and then re-enable *AutoDoc*.

Third, your screens may be mangled by the **SEARCHING...** and **LOADING** prompts the computer displays. Also, if your screen depends on being in lowercase and uppercase mode, you must be in that mode before loading the program. Otherwise, the screen will look funny. *AutoDoc* cannot accommodate for this, so you must be aware of it. Here, consistency in your documentation files helps.

AutoDoc is compatible with *ML Macros* (May 1993), making these programs an unbeatable pair.

Cameron Kaiser, the author of ML Macros (May 1993) is a freshman at the University of California at San Diego. His interests include Pink Floyd, basketball, rock music, and his aging but still useful 128. He lives in La Mesa, California.

Multiraid

Tom Dively

Multiraid is a public domain program that helps teach multiplication. It requires a joystick plugged into port 2.

It does not have its own separate documentation. For more information about this program, read "PD Picks" on this disk.

Putt-Putt

Bill Britton

Putt-Putt, one of the public domain programs featured in this month's "PD Picks," is a fun and challenging game that simulates a round of miniature golf. Documentation is included within the game.

Putt-Putt is incompatible with the *Gazette* menu program. To run the game, reboot your computer, insert side 2, and load it as you would any BASIC program.

Type **LOAD"PUTT PUTT",8** and then press Return.

Gazette Disk Bonus

SIGN UP A FRIEND AND GET A FREE GAZETTE DISK!

That's right. Sign up a friend to a one year's subscription to *Gazette Disk* and we'll extend your subscription by an extra issue.

Sign up two friends, get two free disks. If your friend signs up for two years, we'll see that you get two extra disks.

You probably know many Commodore owners who do not subscribe to *Gazette Disk*. Show this disk to them and see if they don't agree that 12 issues for only \$29.95 is a great deal.

User group members, you can assign the extra disk to your group and any other subscriber of your choice. Impress your friends with your generosity!

Call (800) 727-6937 and use your VISA or MasterCard, or just say, "Bill me!"

If you prefer, just print out this page and send a check or money order for \$29.95 (\$42.75 in Canada, \$39.95 in other countries) to

COMPUTE GAZETTE DISK
P.O. Box 3244
Harlan, IA 51593-3244

Key: HSSF3
New subscriber's
Name:
Address:
City
State/Province:
Country

☐ Payment enclosed ☐ Bill me

Charge ☐ VISA ☐ MasterCard

Account #
Expiration date

Signature

Add the free disk to this subscription: Name:
Address:
City:
State/Province:
Country:

This price is valid through March 15, 1994.

SUBSCRIBE NOW
TO THE ALL NEW

GAZETTE

PRICES
HELD
OVER

DISK

GET 12 MONTHLY ISSUES FOR
ONLY \$29.95 (\$42.75 IN CANADA)

CALL (800) 727-6937
OR WRITE
GAZETTE, BOX 3244, HARLAN, IA 51537

DO YOU HAVE A
COMMODORE 64 OR 128
PRODUCT TO SELL?

ADVERTISE IT HERE ON
THE NEW GAZETTE DISK

CREATE YOUR OWN
OR USE "SCREEN THING"
ON THIS DISK TO DESIGN
YOUR OWN COLORFUL AD

CALL LUCILLE DENNIS FOR DETAILS

(707) 451-8209 VOICE
(707) 451-4269 FAX

REMEMBER ...

GAZETTE HAS CHANGED ITS FORMAT,
BUT IT STILL WANTS TO PURCHASE
YOUR BEST PROGRAMS AND ARTICLES.

SEND A DISK THAT CONTAINS YOUR
PROGRAM AND DOCUMENTATION, PLUS
A HARD COPY OF THE DOCS TO

GAZETTE SUBMISSION
324 W. WENDOVER AVE., STE. 200
GREENSBORO, NC 27408

NOW THAT WE'RE ON DISK, WE CAN
USE YOUR LARGER PROGRAMS!

COMMODORE 64/128

PUBLIC DOMAIN
SHAREWARE PROGRAMS
CALL OR WRITE FOR FREE
DESCRIPTIVE CATALOG OF
COMMODORE 64/128 PUBLIC
DOMAIN & SHAREWARE SOFT-
WARE OR SEND \$2.00 FOR A
DESCRIPTIVE CATALOG AND
SAMPLE DISK. ADULT
SOFTWARE LISTING AVAILABLE
TO THOSE OF LEGAL AGE.

DISKS O'PLENTY, INC.
8362 PINES BLVD., SUITE 270
PEMBROKE PINES, FL 33024
(305) 963-7750

B L U E R E A D E R 4.1

Read/Write MS-DOS disks on your C64/128

Transfers: Word Processing, Text, ASCII
& Binary files between Commodore & IBM
MS-DOS disks.

Requires: 1571 or 1581 Disk Drive or
use CMD's FD-2000/4000 to read/write
special formatted 1.44 MS-DOS disks.

Optional ASCII translation, converts:
Commodore ASCII and Screen Code files
to/from IBM (MS-DOS) Standard ASCII.

Includes both C64 & C128 programs:

BBR 128/64 Version 4.1 only \$35

Send Check, Money Order, or COD to:
SUBMAP Software +++ Ph. 1-219-724-3900
115 Belmont Road; Decatur, IN 46733
<SPACE>=More <+>=Gazette Menu

B i b l e S e a r c h 3.2

Software for Life! - 2Cor 10:5 Rom 10:9

Entire Old & New Testament Bible text
on 4-1541/71 or 2-1581 disks.

Complete Concordance with 700,000+ words
indexed on 2-1541/71 or 1-1581 disks.

Incredible 5 sec. look-up time per/word,
per/disk. [Very Fast & Easy-to-Use.]

Auto spell checking of 12,000+ words,
Search using: Single & Multiple word(s)
ie. Find every usage of Jesus in 5 sec.

Available on 7-1541 or 4-1581 disks:
Includes disk case, and printed manual

KJV \$39.95 / NIV \$49.95 / Both \$75.00

Send Check, Money Order, or COD to:
SUBMAP Software +++ Ph. 1-219-724-3900
115 Belmont Road; Decatur, IN 46733
<SPACE>=More <+>=Gazette Menu



Parsec Ad

Get a free catalog from the company
that has been providing Commodore
support since 1986!

We produce Commodore books, software,
hardware, and magazines. Find out
what we have to offer by sending your
name and address to:

Parsec, Inc.
PO Box 111
Salem, MA 01970-0111
USA

For customers outside of the US or
Canada, to receive our catalog by
Airmail, you should include (6)
STAMPED I.R.C.s otherwise it will be
mailed by surface mail.

FGM Connection Ad

The FUN GRAPHICS MACHINE (FGM) is a multi-featured graphics manipulation program for the C64. You can import graphics, add text, and manipulate pixels until your fingers grow numb. Then print your creations with the multi-featured PRINTER section.

Create forms, newsletters, cards, custom labels, letterhead; the only limit is your imagination.

The FUN GRAPHICS MACHINE.....\$24.95
(includes 140 page manual)

FGM KEYBOARD OVERLAY.....\$ 3.50
(specify C64, C128, SX-64)

FGM SUPPORT DISKS:

FGM FONTS-1 (over 75 fonts).....\$ 5.00

FGM GRAPHICS-1 (250 PS images)..\$ 5.00

FGM CLIP ART-1 (four sides).....\$ 8.00

FGM CLIP ART-2 (two sides).....\$ 5.00

FGM CLIP ART-3 (with ClipCat)...\$ 5.00

FGM CALENDAR TEMPLATE DISK.....\$ 5.00

FGM DEMO DISK.....\$ 2.00

U.S. add \$3.50 for S&H (Priority).
Canada/Mexico add \$4.50 S&H (air).
Other overseas add \$8.50 S&H (air).
U.S. funds only.

The FGM CONNECTION
P.O. BOX 2206
ROSEBURG OR 97470

Phone (503)-673-2234